Firebird White Paper

Requirements for a virtual machine with strong Firebird performance

Holger Klemt, February 2024

The following factors are decisive for the host configuration

1. Single dedicated VM, no other I/O-relevant VMs should run on the same hardware/CPU.
2. CPU selection: Important: If possible, approx. 5 GHz maximum speed available and activated on at least one core.
3. CPU selection: How many cores the CPU has is less important, unless a large number of users are connected to Firebird with a very high read/write load. In single dedicated VM operation, all cores available in the CPU hardware should be activated in the VM.
4. Data carrier connection: Maximum IOPS with minimum latency, preferably high-speed NVMe directly on the mainboard where the host CPU is running.
5. Operation on external storage is rarely fast enough.

Why is that the case?

Firebird is an open-source database system that is characterised by extreme stability and very high speed. A feature of the architecture, known as Careful Write, is extremely important for this speed.

All operations written to the database file do not occur in an external transaction log area, which in the event of an error then locks the database after a reboot until the (hopefully still complete) transaction log in the database file has been processed, before new client connections can be resumed (older readers may remember the chkdsk execution required at Windows or DOS system startup before the computer could be used again). This is exactly the procedure used by most other database manufacturers.

With Firebird, on the other hand, the Firebird server process ensures that all writing operations in the database file are written in an exactly defined sequence, so that the contents of this file, even in a worst case situation (unexpected reboot or other crash), starts immediately after the operating systems restarts, so that the Firebird server process can be opened as a database, without any repair being necessary, and is immediately available for all read/write operations. Everything that has reached the transaction status commit in the database is then also contained in the database file with 100% certainty.

Terms from other worlds such as "repair index" do not exist in the Firebird world, unless the operating system interferes with the Careful Write sequence by making unwanted changes to the write sequence. Unfortunately, this cannot be ruled out with every operating system and, in particular, a virtual host configuration with external storage often offers optimisations based on better performance, the effects of which are only experienced in the worst case scenario.

Fundamentals

The Firebird server process uses the usual Forced Writes attribute, to ensure that write operations are transmitted to the operating system in the correct order, and that the server process waits for the successful completion message from the operating system, before the client application receives the message that the operation was successful. Caching is explicitly not desirable, because if there is an error, the user should not get the impression that their data has been written successfully.

The Firebird server process has organised the entire database file in pages for these write operations.

While information such as metadata, transaction statuses and page content directories can be found at the front of the file, relevant data pages and index pages for the table can be found further back in the database file.

With a tool such as the Process Monitor from syinternals.com, you can, for example, observe the exact write sequence on a Windows server. The positions are therefore very widely distributed in the database file in relatively small blocks; front/back/front/back... etc. And the more clients working simultaneously on the database, the more frequently this must be coordinated in the file.

The contents of the database are stored in the RAM by the database software itself in order to minimise read accesses. From Firebird version 3 onwards, a shared cache for all clients is also kept in the RAM, therefore making significantly fewer read accesses with a much higher number of clients. From Firebird 3 onwards, each core is also used effectively, but only if the number of active clients making real SQL commands corresponds to the number. A higher number is automatically distributed efficiently. However, it is clearly better if, for example, 8 cores on a server can each reach 4 GHz if required, than if you have 64 cores but none of them delivers more than 2 GHz. This applies in particular because all cores have to share the data paths in the CPU socket and memory-intensive operations otherwise quickly lead to serialisation in multicore operation, even though the total CPU cores could actually be capable of more.

The speed at which the server can then process the data internally in the CPU until it is returned to the RAM is explicitly dependent on the maximum clock rate of the processor. As there is little real CPU load involved, but instead very intensive byte shifting, it can be assumed that important operations within the database, which do not yet pass to the external data carrier, take twice as long at a rate of 2 GHz compared to a CPU that processes it at 4 GHz.

Difference to other systems

In contrast to a transaction log-based system, which only has to keep a transaction log as a large single file and, in the event of a system crash, generates a functional database from this and the data in the remaining database file before it can be used again, the I/O behaviour of Firebird is completely different, with the highest priority on reliability and immediate usability even after a system crash.

A transaction log-based system only maintains one large file of any length, so to speak, which it then at some point incorporates back into the database file, in the meantime storing the real content of the database file in its memory.

The behaviour of this architecture can be compared to copying a large 5 GB file from one path to another. With Firebird, on the other hand, the profile is more like a parallel copy process that simultaneously attempts to copy 100,000 small files to 10,000 different paths.

Under Windows you shouldn't really compare this in the NTFS file system, because that pushes every Windows machine to its performance limits anyway; but because the Firebird server manages the file content itself, without creating individual Windows files, this part of the problem is negligible. And it is also very important that the Firebird server, with the forced writes request, instructs the operating system to perform the write operation immediately and report success before the next one comes. Hence the important latency requirement.

IOPS and latency

Suitable modern server hardware, such as our usually non-virtualised IFS servers, achieve read/write performance of 4-5 GB per second on NVMe drives directly on the mainboard, and even in benchmarks with smaller packages of 16 or 32 k multithreaded, this is still 1-1.5 GB per second.

The ATTO Disk Benchmark is a good tool for this. For effective operation with Firebird, however, we recommend the IBExpert Benchmark, which can be downloaded free of charge from ibexpert.com/benchmark.

Our IFS servers are equipped with hardware that can provide the software with 300,000 – 450,000 IOPS with virtually no latency and thus achieve these values.

Worst case scenarios

According to the manufacturer, even the large manufacturers of transaction-based database systems, extensive administrative knowledge is required, not only in the event of an error, in order to reliably restart the system even with a large number of users and data if there were specific errors. Or, as it was the case with a customer project involving a test of the Oracle database: following a database crash simply crash the server again during the repair process. The test at a large press agency resulted in a minimum waiting time of 30 minutes until the system was available again.

Another example that is often cited is a system that is installed in American tanks, and the position of the next artillery attack indicates where you should not be standing, even if you are inside an American tank. This is based on the technology of the Firebird predecessor InterBase and is characterised by the fact that even after the unavoidable electromagnetic pulse, which occurs in a tank when a grenade is fired at the front, all computers crash due to antenna effects on the conductor path, and the only question is when will data be available again after the reboot. Having to wait until the computer has rebooted before firing the next shot is certainly a tactical problem.

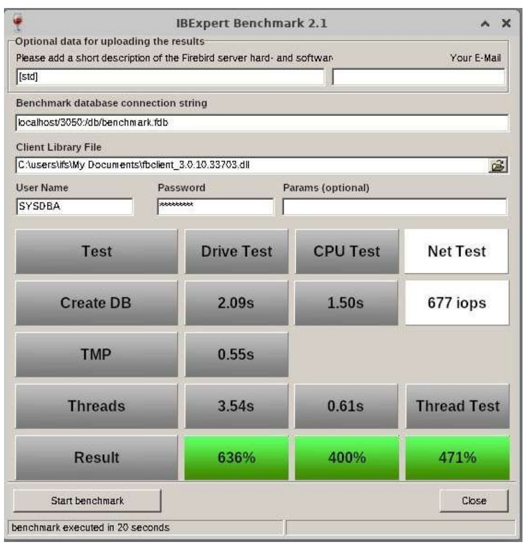
Conclusion

When a server used for virtualisation with suitable hardware as described above, delivers very good benchmark values without a virtual host around it, the final performance is then only influenced by the VM host software solution and the configuration selected. However, if the hardware is unsuitable, even the best customisation of the VM host configuration will not help to bring the system anywhere near a suitable performance level.

Even the storage used in a large German hospital, which had several petabytes of memory and was capable of handling all caching processes with built-in RAM/SSD/NVME technology, was clearly too slow to be used as storage for a Firebird VM. It didn't help that the storage cost around 1 million euros. Less IOPS-heavy or latency-critical applications could be used very well with the system. However, this system was unsuitable for use as a Firebird server and any 100 Euro Sata SSD installed directly in the host hardware would have provided significantly better performance.

It is therefore important to select suitable hardware independently of virtualisation. You can test whether this is suitable or not yourself at any time under Windows using the IBExpert Benchmark. If the result is clearly in the green range, you can use this hardware, if the values are below 100%, then preferably not. We have referenced a server that we supplied to customers 13 years ago with 100%. Our current servers achieve 400%-650% for a budget of approx. 5,000 euros per server.

https://ibexpert.com/benchmark



Do you have any questions?

IBExpert GmbH
Oldenburger Strasse 233
26203 Wardenburg
Germany
www.ibexpert.com
sales@ibexpert.biz
Telefon: +49 4407 3148770